

# Maximum likelihood decoding for multilevel channels with gain and offset mismatch

Simon R. Blackburn  
 Department of Mathematics  
 Royal Holloway University of London  
 Egham, Surrey TW20 0EX  
 United Kingdom

September 10, 2015

## Abstract

K.A.S. Immink and J.H. Weber recently defined and studied a channel with both gain and offset mismatch, modelling the behaviour of charge-leakage in flash memory. They proposed a decoding measure for this channel based on minimising Pearson distance (a notion from cluster analysis). The paper derives a formula for maximum likelihood decoding for this channel, and also defines and justifies a notion of minimum distance of a code in this context.

## 1 Introduction

We begin by defining some notation. Let  $n$  be an integer,  $n \geq 3$ . All our vectors will have length  $n$ , and will have entries in the real numbers  $\mathbb{R}$ . For a vector  $\mathbf{x}$ , we write  $x_i$  for the  $i$ th entry of  $\mathbf{x}$ , we write

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n x_i$$

for the mean of  $\mathbf{x}$  and we write

$$\sigma_{\mathbf{x}} = \sqrt{\sum_{i=1}^n (x_i - \bar{\mathbf{x}})^2}$$

for the (unnormalised) standard deviation of  $\mathbf{x}$ . We write  $\mathbf{1}$  for the all-one vector of length  $n$ , and call any scalar multiple of  $\mathbf{1}$  a constant vector. For vectors  $\mathbf{u}$  and  $\mathbf{v}$  that are not constant vectors, the *Pearson correlation coefficient*  $\rho_{\mathbf{u},\mathbf{v}}$  is defined by

$$\rho_{\mathbf{u},\mathbf{v}} = \frac{\sum_{i=1}^n (u_i - \bar{\mathbf{u}})(v_i - \bar{\mathbf{v}})}{\sigma_{\mathbf{u}}\sigma_{\mathbf{v}}}.$$

Finally, the *Pearson distance*  $\delta_{\text{Pearson}}(\mathbf{u}, \mathbf{v})$  between vectors  $\mathbf{u}$  and  $\mathbf{v}$  is defined to be

$$\delta_{\text{Pearson}}(\mathbf{u}, \mathbf{v}) = 1 - \rho_{\mathbf{u},\mathbf{v}}.$$

Since  $\rho_{\mathbf{u},\mathbf{v}}$  lies between  $-1$  and  $1$ , the Pearson distance lies between  $0$  and  $2$ . Both Pearson distance and Pearson correlation are well-known concepts in the area of cluster analysis.

The channel considered by Kees A. Schouhamer Immink and Jos H. Weber [3] is defined as follows. If the vector  $\mathbf{x}$  is sent through the channel, the channel outputs the received vector  $\mathbf{r}$  where

$$\mathbf{r} = a(\mathbf{x} + \boldsymbol{\nu}) + b\mathbf{1}.$$

Here  $a$  (the *gain*) and  $b$  (the *offset*) are unknown real numbers, with  $a > 0$ , and

$$\boldsymbol{\nu} = (\nu_1, \nu_2, \dots, \nu_n)$$

where the  $\nu_i$  are independently normally distributed with mean  $0$  and standard deviation  $\sigma$ .

The channel is motivated by the properties of flash memory. We give some basic details of this setting here; see [1, 9] for more detailed introductions, and see (for example) [6, 7, 8] for another approach to modelling the problem using rank modulation codes. Flash memory is made up of an array of floating-gate transistors, known as flash cells. Data is stored in each cell by varying the charge (equivalently, the voltage) on the cell. In single level cell (SLC) flash memory, each cell stores one bit of information depending on whether the voltage level is zero or non-zero. In more recent multi-level cell (MLC) systems, more information is stored by allowing the cell to be charged at one of several discrete non-zero voltage levels. The vector  $\mathbf{x}$  corresponds to the voltages we wish to store in a block of  $n$  cells, so  $x_i$  is the voltage we wish to store in the  $i$ th cell. We cannot hope to initialise a cell with the exact voltage we wish: the errors in this process give rise to the error term  $\boldsymbol{\nu}$ .

Over time, the voltage in each cell drops due to charge leakage. We assume that the function that gives this voltage change is unknown, but is affine and is independent of which cell in the block we are examining. The unknown coefficients  $a$  and  $b$  specify this function; the coefficient  $a$  is positive since charge leakage is monotonic increasing: the more charge we have initially, the more we have after leakage. The received vector  $\mathbf{r}$  thus models the set of voltages we retrieve from a block of cells we have initialised with voltages corresponding to  $\mathbf{x}$ .

We note that the channel does not model some aspects of flash memory: intercell coupling (where the charge on one cell influences the charge on neighbouring cells) is not modelled in any way; nor is the possibility that the magnitude of the error in the charging process depends on the charge in some way. Nevertheless, the channel is very natural and captures key properties of the process of retrieving data from flash memory.

Immink and Weber assume that the vectors  $\mathbf{x}$  lie in some finite subset  $C$  of  $\mathbb{R}^n$ . (In fact, they assume that  $C \subseteq \{0, 1, \dots, q-1\}^n$  for some fixed integer  $q$ .) This corresponds to the fact that we initialise each cell with one of a finite discrete set of voltages. To ensure unique decoding in the absence of noise, they assume that if  $\mathbf{x} \in C$  then no other codeword  $\mathbf{y} \in C$  has the form  $\mathbf{y} = a\mathbf{x} + b\mathbf{1}$  for real numbers  $a$  and  $b$  with  $a$  positive. They also assume that no constant vector lies in  $C$ . This makes the Pearson distance between any pair of vectors in  $C$  well-defined; see Section 6 for additional motivation for this assumption. Weber, Immink and Blackburn [5] have studied maximal codes  $C \subseteq \{0, 1, \dots, q-1\}^n$  with these properties.

A decoder based on Pearson distance is proposed in this setting in [3]. So we decode a received vector  $\mathbf{r}$  as  $\hat{\mathbf{x}}$ , where  $\hat{\mathbf{x}} \in C$  minimises  $\delta_{\text{Pearson}}(\mathbf{r}, \hat{\mathbf{x}})$ . One motivation for this choice is that Pearson distance behaves well with respect to an affine charge-leakage function, since

$$\delta_{\text{Pearson}}(\mathbf{r}, \hat{\mathbf{x}}) = \delta_{\text{Pearson}}(a\mathbf{r} + b\mathbf{1}, a\hat{\mathbf{x}} + b\mathbf{1}).$$

Pearson distance has a natural geometric meaning: see Section 6 for a brief discussion.

In this paper, we derive a maximum likelihood decoding function for the channel in [3], and compare a decoder based on this function with a decoder based on minimising Pearson distance. We also propose and justify a notion of minimum distance for codes used with this channel.

We should emphasise that the model makes no assumptions on the distribution of the unknown (‘nuisance’) parameters  $a$  and  $b$ : if we know something

about these distributions, other decoding methods might be appropriate. For example, if  $a$  is known to be very close to 1, then decoding based on minimising Euclidean distance is sensible; Immink and Weber [4] have proposed a decoder based on minimising a weighted sum of Euclidean and Pearson distances in some situations.

The remainder of the paper is structured as follows. Section 2 sets up notation, and contains some preliminary lemmas. In Section 3 we show how to achieve Maximum Likelihood Decoding for this channel. Pearson distance is not the measure to use for Maximum Likelihood decoding, but is often a good approximation to it: simulations show comparable performance between both MLD and Pearson decoders. Section 4 defines and justifies a minimum distance measure for codes designed for the channel. In Section 5, we give some results of simulations that compare the approach in [3] with the one taken here. Finally, Section 6 provides some comments on various aspects of the model in [3].

## 2 Preliminaries

This section contains notation that will be used in the remainder of this paper. Some simple facts, which will often be used without further comment, are also stated.

We define  $\|\mathbf{u}\|$  to be the Euclidean length of  $\mathbf{u} \in \mathbb{R}^n$ , and we define  $\delta(\mathbf{u}, \mathbf{v})$  to be the Euclidean distance between  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ .

Define the subspace  $Z$  of  $\mathbb{R}^n$  by

$$\begin{aligned} Z &= \{\mathbf{x} \in \mathbb{R}^n : \bar{\mathbf{x}} = 0\} \\ &= \left\{ (x_1, x_2, \dots, x_n) \in \mathbb{R}^n : \sum_{i=1}^n x_i = 0 \right\}. \end{aligned}$$

Let  $\zeta : \mathbb{R}^n \rightarrow Z$  be defined by

$$\zeta(\mathbf{x}) = \mathbf{x} - \bar{\mathbf{x}}\mathbf{1}.$$

We can think of  $\zeta$  as a ‘normalisation’, applying an offset to a vector so that it has mean zero. Using  $\zeta$  allows the formulas given in the introduction to be expressed in a more geometric way. We now give more details. We see that

$$\sigma_{\mathbf{u}} = \|\zeta(\mathbf{u})\|. \tag{1}$$

We write  $\langle \mathbf{x}, \mathbf{y} \rangle$  for the standard inner product (the dot product) of  $\mathbf{x}$  and  $\mathbf{y}$ . So

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^n x_i y_i.$$

Since  $\langle \mathbf{x}, \mathbf{y} \rangle = \|\mathbf{x}\| \|\mathbf{y}\| \cos \theta$  where  $\theta$  is the angle between  $\mathbf{x}$  and  $\mathbf{y}$ , we see that

$$\rho_{\mathbf{u}, \mathbf{v}} = \frac{\langle \zeta(\mathbf{u}), \zeta(\mathbf{v}) \rangle}{\sigma_{\zeta(\mathbf{u})} \sigma_{\zeta(\mathbf{v})}} \quad (2)$$

$$\begin{aligned} &= \frac{\|\zeta(\mathbf{u})\| \|\zeta(\mathbf{v})\| \cos \theta}{\|\zeta(\mathbf{u})\| \|\zeta(\mathbf{v})\|} \\ &= \cos \theta, \end{aligned} \quad (3)$$

where  $\theta$  is the angle between  $\zeta(\mathbf{u})$  and  $\zeta(\mathbf{v})$ .

Finally, we note that  $\zeta(\zeta(\mathbf{u})) = \zeta(\mathbf{u})$ , that  $\zeta(\mathbf{u} + \mathbf{v}) = \zeta(\mathbf{u}) + \zeta(\mathbf{v})$  and that  $\sigma_{\zeta(\mathbf{u})} = \sigma_{\mathbf{u}}$ .

### 3 Maximum likelihood decoding

This section provides a proof of the following theorem:

**Theorem 1.** *A maximum likelihood decoder decodes a received vector  $\mathbf{r}$  to the codeword  $\hat{\mathbf{x}}$  which minimises  $\ell_{\mathbf{r}}(\hat{\mathbf{x}})$ , where*

$$\ell_{\mathbf{r}}(\hat{\mathbf{x}}) = \begin{cases} \sigma_{\hat{\mathbf{x}}}^2 (1 - \rho_{\mathbf{r}, \hat{\mathbf{x}}}^2) & \text{when } \rho_{\mathbf{r}, \hat{\mathbf{x}}} > 0, \\ \sigma_{\hat{\mathbf{x}}}^2 & \text{otherwise.} \end{cases} \quad (4)$$

Before proving this theorem, we provide a geometrical interpretation for the formula (4). For a non-zero vector  $\mathbf{r} \in \mathbb{R}^n$ , define

$$\begin{aligned} U_{\mathbf{r}} &= \{a'\mathbf{r} + b'\mathbf{1} \mid a', b' \in \mathbb{R}\}, \text{ and} \\ U_{\mathbf{r}}^+ &= \{a'\mathbf{r} + b'\mathbf{1} \mid a', b' \in \mathbb{R}, a' > 0\}. \end{aligned}$$

So  $U_{\mathbf{r}}$  is a subspace, and  $U_{\mathbf{r}}^+$  is a half-subspace, of  $\mathbb{R}^n$ . For a vector  $\mathbf{r} \in \mathbb{R}^n$  we write  $R_{\mathbf{r}}$  for the ray from the origin in the direction of  $\mathbf{r}$ , so

$$R_{\mathbf{r}} = \{a'\mathbf{r} : a' \in \mathbb{R}, a' > 0\}.$$

**Lemma 2.** Let  $\mathbf{r}$  and  $\hat{\mathbf{x}}$  be vectors in  $\mathbb{R}^n$ . Let  $d_1$  be the Euclidean distance between  $\hat{\mathbf{x}}$  and  $U_{\mathbf{r}}^+$ . Let  $d_2$  be the Euclidean distance between  $\zeta(\hat{\mathbf{x}})$  and  $R_{\zeta(\mathbf{r})}$ . Then

$$d_1^2 = d_2^2 = \ell_{\mathbf{r}}(\hat{\mathbf{x}}).$$

*Proof.* We start by proving that  $d_1 = d_2$ . Let  $\mathbf{u} = a'\zeta(\mathbf{r}) = a'\mathbf{r} - a'\bar{\mathbf{r}}\mathbf{1} \in R_{\zeta(\mathbf{r})}$ . Then

$$\delta(\zeta(\hat{\mathbf{x}}), \mathbf{u}) = \delta(\hat{\mathbf{x}} - \bar{\hat{\mathbf{x}}}\mathbf{1}, \mathbf{u}) = \delta(\hat{\mathbf{x}}, \mathbf{u} + \bar{\hat{\mathbf{x}}}\mathbf{1}),$$

and  $\mathbf{u} + \bar{\hat{\mathbf{x}}}\mathbf{1} = a'\mathbf{r} + (\bar{\hat{\mathbf{x}}} - \bar{\mathbf{r}})\mathbf{1} \in U_{\mathbf{r}}^+$ . So  $d_1 \leq d_2$ .

Let  $\mathbf{u} = a'\mathbf{r} + b'\mathbf{1} = a'\zeta(\mathbf{r}) + (b' + \bar{\mathbf{r}})\mathbf{1} \in U_{\mathbf{r}}^+$ . Then

$$\delta(\hat{\mathbf{x}}, \mathbf{u}) = \delta(\zeta(\hat{\mathbf{x}}), \mathbf{u} - \bar{\hat{\mathbf{x}}}\mathbf{1}) = \delta(\zeta(\hat{\mathbf{x}}), a'\zeta(\mathbf{r}) + (b' + \bar{\mathbf{r}} - \bar{\hat{\mathbf{x}}})\mathbf{1}) \geq \delta(\zeta(\hat{\mathbf{x}}), a'\zeta(\mathbf{r})),$$

since  $\zeta(\hat{\mathbf{x}}), a'\zeta(\mathbf{r}) \in Z$  and since  $\mathbf{1}$  is orthogonal to  $Z$ . Since  $a'\zeta(\mathbf{r}) \in R_{\zeta(\mathbf{r})}$ , we see that  $d_2 \leq d_1$ . Hence  $d_1 = d_2$ .

We now prove that  $d_2^2 = \ell_{\mathbf{r}}(\hat{\mathbf{x}})$ . There are two cases, depending on whether or not the closest point  $P$  to  $\zeta(\hat{\mathbf{x}})$  on the line generated by  $\zeta(\mathbf{r})$  lies in the ray  $R_{\zeta(\mathbf{r})}$ : see Figure 1. The first case, when  $P$  lies on the ray, happens if and only if  $\langle \zeta(\hat{\mathbf{x}}), \zeta(\mathbf{r}) \rangle > 0$ . This happens exactly when  $\rho_{\hat{\mathbf{x}}, \mathbf{r}} > 0$ , by (2). In this case,

$$d_2^2 = \|\zeta(\hat{\mathbf{x}})\|^2 \sin^2 \theta = \|\zeta(\hat{\mathbf{x}})\|^2 (1 - \cos^2 \theta) = \sigma_{\hat{\mathbf{x}}}^2 (1 - \rho_{\mathbf{r}, \hat{\mathbf{x}}}^2),$$

where  $\theta$  is the angle between  $\zeta(\hat{\mathbf{x}})$  and  $\zeta(\mathbf{r})$ , by (1) and (3). In the second case, when  $\langle \zeta(\hat{\mathbf{x}}), \zeta(\mathbf{r}) \rangle \leq 0$ , the distance between  $\zeta(\hat{\mathbf{x}})$  and the ray  $R_{\zeta(\mathbf{r})}$  is given by the distance from  $\zeta(\hat{\mathbf{x}})$  to the origin. So

$$d_2^2 = \|\zeta(\hat{\mathbf{x}})\|^2 = \sigma_{\hat{\mathbf{x}}}^2,$$

by (1). This establishes the lemma.  $\square$

*Proof of Theorem 1.* Since the components of  $\boldsymbol{\nu}$  are picked independently according to a normal distribution with mean 0 and standard deviation  $\sigma$ , each value of  $\boldsymbol{\nu}$  is associated with the value of the corresponding normal Probability Density Function  $f(\boldsymbol{\nu})$ , where

$$f(\boldsymbol{\nu}) = \prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} \exp(-\nu_i^2/(2\sigma^2)).$$

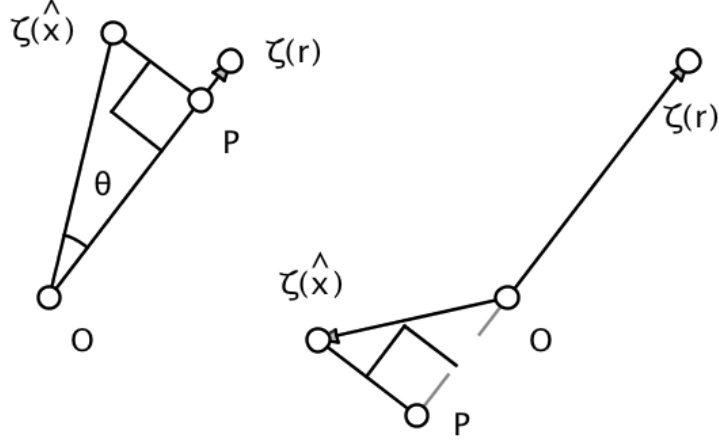


Figure 1: The distance of a point to a ray: two cases

For vectors  $\mathbf{r}$  and  $\hat{\mathbf{x}}$ , define

$$L_{a,b}(\hat{\mathbf{x}} \mid \mathbf{r}) = f((\mathbf{r} - b\mathbf{1})/a - \hat{\mathbf{x}}).$$

This is the likelihood of  $\hat{\mathbf{x}}$  given  $\mathbf{r}$  when  $a$  and  $b$  are fixed, since  $\boldsymbol{\nu} = (\mathbf{r} - b\mathbf{1})/a - \hat{\mathbf{x}}$  in this case.

In maximum likelihood decoding, we decode a received vector  $\mathbf{r} = a(\mathbf{x} + \boldsymbol{\nu}) + b\mathbf{1}$  as  $\hat{\mathbf{x}} \in C$ , where  $\hat{\mathbf{x}}$  is the codeword that maximises

$$\begin{aligned} \max_{a,b \in \mathbb{R}, a > 0} L_{a,b}(\hat{\mathbf{x}} \mid \mathbf{r}) &= \max_{a,b \in \mathbb{R}, a > 0} f((\mathbf{r} - b\mathbf{1})/a - \hat{\mathbf{x}}) \\ &= \max_{a',b' \in \mathbb{R}, a' > 0} f(a'\mathbf{r} + b'\mathbf{1} - \hat{\mathbf{x}}), \end{aligned}$$

where  $a' = 1/a$  and  $b' = b/a$ . The logarithm function is strictly increasing on the positive real numbers, and  $f$  is a positive function. So equivalently we want to find  $\hat{\mathbf{x}} \in C$  that maximises  $\max_{a',b' \in \mathbb{R}, a' > 0} \log f(a'\mathbf{r} + b'\mathbf{1} - \hat{\mathbf{x}})$ . But

$$\log f(a'\mathbf{r} + b'\mathbf{1} - \hat{\mathbf{x}}) = -n \log(\sigma\sqrt{2\pi}) - \frac{1}{2\sigma^2} \sum_{i=1}^n (a'r_i + b' - \hat{x}_i)^2.$$

Since  $-n \log(\sigma\sqrt{2\pi})$  is a constant (in other words, independent of  $\hat{\mathbf{x}}$  and  $\mathbf{r}$ ), and since  $\frac{1}{2\sigma^2}$  is a positive constant, we see that a maximum likelihood

decoder finds a codeword  $\hat{\mathbf{x}}$  that *minimises*

$$\min_{a', b' \in \mathbb{R}, a' > 0} \sum_{i=1}^n (a' r_i + b' - \hat{x}_i)^2,$$

which is the square of the Euclidean distance between  $U_{\mathbf{r}}^+$  and  $\hat{\mathbf{x}}$ . But, by Lemma 2, this is exactly the same as minimising the function  $\ell_{\mathbf{r}}(\hat{\mathbf{x}})$ , as required.  $\square$

We describe techniques to reduce the amount of computation the maximum likelihood decoder needs. Firstly, the value  $\sigma_{\hat{\mathbf{x}}}^2$  can be precomputed for all codewords  $\hat{\mathbf{x}} \in C$ . Secondly, for codes such as 2-constrained codes [3] that are preserved under permuting their coordinates, we can significantly reduce the number of codewords we need to consider by making the following observations. The value of  $\sigma_{\hat{\mathbf{x}}}$  is not changed if we permute the coordinates of  $\hat{\mathbf{x}}$ , and the value of  $\rho_{\mathbf{r}, \hat{\mathbf{x}}}$  is maximised when we permute the coordinates of  $\hat{\mathbf{x}}$  to have the same order as the coordinates of  $\mathbf{r}$ . So we may use the ‘composition code’ decomposition technique from [3, Section IV.B] to decode more efficiently, only storing codewords that are in sorted order. Finally, we observe that 2-constrained codes  $C$  have the property that whenever  $\mathbf{x} \in C$  then its *complement*  $\mathbf{y} = (q-1)\mathbf{1} - \mathbf{x}$  also lies in  $C$ . We note that  $\zeta(\mathbf{x}) = -\zeta(\mathbf{y})$  and so we find that  $\sigma_{\mathbf{x}} = \sigma_{\mathbf{y}}$  and  $\rho_{\mathbf{r}, \mathbf{x}} = -\rho_{\mathbf{r}, \mathbf{y}}$  for any non-constant received word  $\mathbf{r}$ . So for codes which are closed under taking complements, we only need to store one codeword from each pair  $\{\mathbf{x}, \mathbf{y}\}$ . If we do this, we search for a codeword  $\hat{\mathbf{x}}$  that minimises  $\sigma_{\hat{\mathbf{x}}}^2(1 - \rho_{\mathbf{r}, \hat{\mathbf{x}}}^2)$ ; we then decode to the complement of  $\hat{\mathbf{x}}$  when  $\rho_{\mathbf{r}, \hat{\mathbf{x}}} < 0$  and decode to  $\hat{\mathbf{x}}$  otherwise. This technique can be combined with the composition code technique above. The technique can also be used with the decoder in [3]: here we find a codeword maximising  $|\rho_{\mathbf{r}, \hat{\mathbf{x}}}|$ , and decode to this codeword if  $\rho_{\mathbf{r}, \hat{\mathbf{x}}} \geq 0$  or to its complement otherwise.

## 4 The distance between codewords

For codewords  $\mathbf{u}, \mathbf{v} \in C$ , we define a (squared) distance measure  $\delta'(\mathbf{u}, \mathbf{v})$  by

$$\delta'(\mathbf{u}, \mathbf{v}) = \begin{cases} \sigma_{\mathbf{u}}^2 \sigma_{\mathbf{v}}^2 (1 - \rho_{\mathbf{u}, \mathbf{v}}^2) / \sigma_{\mathbf{u} + \mathbf{v}}^2 & \text{when } \rho_{\mathbf{u}, \mathbf{v}} > -\min\{\sigma_{\mathbf{v}} / \sigma_{\mathbf{u}}, \sigma_{\mathbf{u}} / \sigma_{\mathbf{v}}\}, \\ \min\{\sigma_{\mathbf{u}}^2, \sigma_{\mathbf{v}}^2\} & \text{otherwise.} \end{cases}$$



Note that  $\delta'(\mathbf{u}, \mathbf{v}) = \delta'(\mathbf{v}, \mathbf{u})$ . Also note that  $\delta'(\mathbf{u}, \mathbf{v})$  depends only on  $\zeta(\mathbf{u})$  and  $\zeta(\mathbf{v})$ , by (1) and (2). Finally, we claim that

$$\sigma_{\mathbf{u}}^2 \sigma_{\mathbf{v}}^2 (1 - \rho_{\mathbf{u}, \mathbf{v}}^2) / \sigma_{\mathbf{u} + \mathbf{v}}^2 \leq \min\{\sigma_{\mathbf{u}}^2, \sigma_{\mathbf{v}}^2\}. \quad (5)$$

To see this, we may verify by routine calculation that

$$\begin{aligned} \sigma_{\mathbf{u}}^2 \sigma_{\mathbf{v}}^2 (1 - \rho_{\mathbf{u}, \mathbf{v}}^2) / \sigma_{\mathbf{u} + \mathbf{v}}^2 &= \frac{\langle \zeta(\mathbf{u}), \zeta(\mathbf{u}) \rangle \langle \zeta(\mathbf{v}), \zeta(\mathbf{v}) \rangle - \langle \zeta(\mathbf{u}), \zeta(\mathbf{v}) \rangle^2}{\langle \zeta(\mathbf{u} + \mathbf{v}), \zeta(\mathbf{u} + \mathbf{v}) \rangle} \\ &= \langle \zeta(\mathbf{u}), \zeta(\mathbf{u}) \rangle - \frac{\langle \zeta(\mathbf{u}), \zeta(\mathbf{u}) + \zeta(\mathbf{v}) \rangle^2}{\langle \zeta(\mathbf{u}) + \zeta(\mathbf{v}), \zeta(\mathbf{u}) + \zeta(\mathbf{v}) \rangle} \\ &\leq \langle \zeta(\mathbf{u}), \zeta(\mathbf{u}) \rangle = \sigma_{\mathbf{u}}^2. \end{aligned} \quad (6)$$

A similar calculation shows that the left hand side of (5) is at most  $\sigma_{\mathbf{v}}^2$ , and so the claim follows.

In this section, we will give a geometric interpretation for  $\delta'(\mathbf{u}, \mathbf{v})$ , and we relate the minimum distance of a code (using this notion of distance) to the error rate of a maximum likelihood decoder.

We note that [3] defines a different distance measure (namely the distance  $d_2(\mathbf{u}, \mathbf{v}) = 2\sigma_{\mathbf{u}}^2(1 - \rho_{\mathbf{u}, \mathbf{v}})$ , which is not symmetrical in  $\mathbf{u}$  and  $\mathbf{v}$ ) to be used to calculate the minimum distance of a code in this context. This distance measure is natural for the decoder in [3]; in Section 5 we briefly compare this measure with the measure above.

**Lemma 3.** *Let  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ . Then  $\delta' = \delta'(\mathbf{x}, \mathbf{y})$  is the largest real number  $\delta'$  with the following property. Let  $B(\mathbf{x}, \delta')$  be the ball in  $Z$  of radius  $\sqrt{\delta'}$  and centre  $\zeta(\mathbf{x})$  (using Euclidean distance). Let  $B(\mathbf{y}, \delta')$  be the ball in  $Z$  of radius  $\sqrt{\delta'}$  and centre  $\zeta(\mathbf{y})$ . Then there is no ray  $R_{\zeta(\mathbf{r})}$  that intersects the interior of both  $B(\mathbf{x}, \delta')$  and  $B(\mathbf{y}, \delta')$ .*

*Proof.* Firstly, suppose that  $\rho_{\mathbf{x}, \mathbf{y}} > -\min\{\sigma_{\mathbf{y}}/\sigma_{\mathbf{x}}, \sigma_{\mathbf{x}}/\sigma_{\mathbf{y}}\}$ . In particular this means that  $\zeta(\mathbf{x}) \neq -\zeta(\mathbf{y})$ , and so  $\zeta(\mathbf{x} + \mathbf{y})$  is a non-zero vector.

The typical situation in this case is drawn in Figure 2. Let  $P$  be the subplane of  $Z$  generated by  $\zeta(\mathbf{x})$  and  $\zeta(\mathbf{y})$ , and let  $K$  be the subplane of vectors orthogonal to  $P$ . Let  $H$  be the hyperplane in  $Z$  generated by  $\zeta(\mathbf{x} + \mathbf{y})$  and  $K$ . We have that  $\zeta(\mathbf{x})$  and  $\zeta(\mathbf{y})$  lie on different sides of  $H$ . The closest point in  $H$  to  $\zeta(\mathbf{x})$  lies in  $P$ , and so lies on the line generated by  $\zeta(\mathbf{x}) + \zeta(\mathbf{y})$ ; the same is true for the closest point to  $\zeta(\mathbf{y})$ . Setting  $\theta$  to be the angle

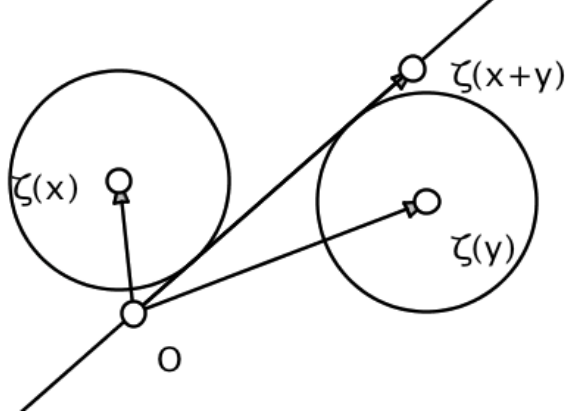


Figure 2: A hyperplane in  $Z$  at equal distance from  $\zeta(\mathbf{x})$  and  $\zeta(\mathbf{y})$

between  $\zeta(\mathbf{x})$  and  $\zeta(\mathbf{x}) + \zeta(\mathbf{y})$ , we find that the squared distance between  $H$  and  $\zeta(\mathbf{x})$  is

$$\begin{aligned}
 \|\zeta(\mathbf{x})\|^2 \sin^2 \theta &= \|\zeta(\mathbf{x})\|^2 - \|\zeta(\mathbf{x})\|^2 \cos^2 \theta \\
 &= \langle \zeta(\mathbf{x}), \zeta(\mathbf{x}) \rangle - \frac{\langle \zeta(\mathbf{x}), \zeta(\mathbf{x}) + \zeta(\mathbf{y}) \rangle^2}{\langle \zeta(\mathbf{x}) + \zeta(\mathbf{y}), \zeta(\mathbf{x}) + \zeta(\mathbf{y}) \rangle} \\
 &= \sigma_{\mathbf{x}}^2 \sigma_{\mathbf{y}}^2 (1 - \rho_{\mathbf{x}, \mathbf{y}}^2) / \sigma_{\mathbf{x} + \mathbf{y}}^2 \text{ by (6)} \\
 &= \delta'.
 \end{aligned}$$

So the interior of  $B(\mathbf{x}, \delta')$  does not intersect  $H$ . Similarly,  $\zeta(\mathbf{y})$  is also at distance  $\delta'$  from  $H$  and so the interior of  $B(\mathbf{y}, \delta')$  does not intersect  $H$ . So the interiors of  $B(\mathbf{x}, \delta')$  and  $B(\mathbf{y}, \delta')$  lie on different sides of a hyperplane, and therefore no ray from the origin intersects them both, as required.

We now show that the value for  $\delta'$  is optimal, by proving that the ray  $R_{\zeta(\mathbf{x} + \mathbf{y})}$  touches the boundaries of both  $B(\mathbf{x}, \delta')$  and  $B(\mathbf{y}, \delta')$ . The nearest point to  $\zeta(\mathbf{x})$  on the line generated by  $\zeta(\mathbf{x} + \mathbf{y})$  is given by

$$\frac{\|\zeta(\mathbf{x})\| \cos \theta}{\|\zeta(\mathbf{x} + \mathbf{y})\|} \zeta(\mathbf{x} + \mathbf{y}) = \frac{\langle \zeta(\mathbf{x}), \zeta(\mathbf{x} + \mathbf{y}) \rangle}{\|\zeta(\mathbf{x} + \mathbf{y})\|^2} \zeta(\mathbf{x} + \mathbf{y}).$$

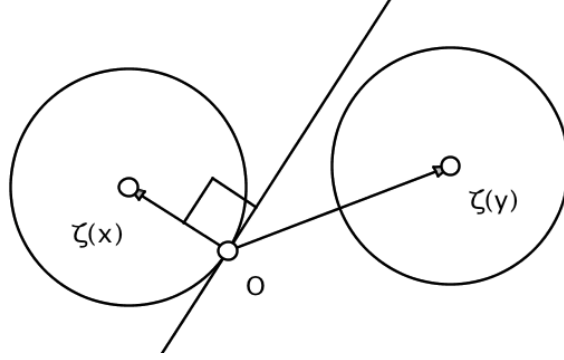


Figure 3: A typical case when  $\|\zeta(\mathbf{x})\| \leq \|\zeta(\mathbf{y})\|$

So  $R_{\zeta(\mathbf{x}+\mathbf{y})}$  touches  $B(\mathbf{x}, \delta')$  if and only if  $\langle \zeta(\mathbf{x}), \zeta(\mathbf{x} + \mathbf{y}) \rangle > 0$ . But

$$\begin{aligned} \langle \zeta(\mathbf{x}), \zeta(\mathbf{x} + \mathbf{y}) \rangle &= \langle \zeta(\mathbf{x}), \zeta(\mathbf{x}) \rangle + \langle \zeta(\mathbf{x}), \zeta(\mathbf{y}) \rangle \\ &= \sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{x}}\sigma_{\mathbf{y}}\rho_{\mathbf{x},\mathbf{y}} \text{ by (1) and (2)} \\ &> \sigma_{\mathbf{x}}^2 - \sigma_{\mathbf{x}}\sigma_{\mathbf{y}}(\sigma_{\mathbf{x}}/\sigma_{\mathbf{y}}) \\ &= 0. \end{aligned}$$

So  $R_{\zeta(\mathbf{x}+\mathbf{y})}$  touches  $B(\mathbf{x}, \delta')$ . The argument that  $R_{\zeta(\mathbf{x}+\mathbf{y})}$  touches  $B(\mathbf{y}, \delta')$  is similar, and uses the fact that  $\rho_{\mathbf{x},\mathbf{y}} > -\sigma_{\mathbf{y}}/\sigma_{\mathbf{x}}$ . This shows that our value for  $\delta'$  is optimal in this case.

We now turn to the case when  $\rho_{\mathbf{x},\mathbf{y}} \leq -\min\{\sigma_{\mathbf{y}}/\sigma_{\mathbf{x}}, \sigma_{\mathbf{x}}/\sigma_{\mathbf{y}}\}$ . See Figure 3 for a typical situation. Without loss of generality, assume that  $\sigma_{\mathbf{x}} \leq \sigma_{\mathbf{y}}$ . So  $\delta' = \sigma_{\mathbf{x}}^2$  and  $\rho_{\mathbf{x},\mathbf{y}} \leq -\sigma_{\mathbf{x}}/\sigma_{\mathbf{y}}$ .

Let  $H = \zeta(\mathbf{x})^\perp$ , so

$$H = \{\mathbf{u} \in Z : \langle \zeta(\mathbf{x}), \mathbf{u} \rangle = 0\}$$

is the hyperplane in  $Z$  of all vectors that are orthogonal to  $\mathbf{x}$ . Clearly the nearest point on  $H$  to  $\zeta(\mathbf{x})$  is the origin, so  $\zeta(\mathbf{x})$  is at distance  $\|\zeta(\mathbf{x})\| = \sigma_{\mathbf{x}}$  from  $H$ . Moreover, all points  $\mathbf{u}$  in the interior of  $B(\mathbf{x}, \delta')$  have  $\langle \zeta(\mathbf{x}), \mathbf{u} \rangle > 0$ . Now let  $\mathbf{u}$  be a point in the interior of  $B(\mathbf{y}, \delta')$ . Then  $\mathbf{u} = \zeta(\mathbf{y}) + \mathbf{v}$ , where

$\|\mathbf{v}\| < \sqrt{\delta'}$  and so

$$\begin{aligned}
\langle \zeta(\mathbf{x}), \mathbf{u} \rangle &= \langle \zeta(\mathbf{x}), \zeta(\mathbf{y}) \rangle + \langle \zeta(\mathbf{x}), \mathbf{v} \rangle \\
&< \langle \zeta(\mathbf{x}), \zeta(\mathbf{y}) \rangle + \|\zeta(\mathbf{x})\| \sqrt{\delta'} \\
&\leq \|\zeta(\mathbf{x})\| \|\zeta(\mathbf{y})\| \rho_{\mathbf{x}, \mathbf{y}} + \|\zeta(\mathbf{x})\| \sigma_{\mathbf{x}} \\
&\leq \sigma_{\mathbf{x}} \sigma_{\mathbf{y}} (-\sigma_{\mathbf{x}} / \sigma_{\mathbf{y}}) + \sigma_{\mathbf{x}}^2 \\
&= 0.
\end{aligned}$$

Thus all points in the interior of  $B(\mathbf{y}, \delta')$  lie on the opposite side of the hyperplane  $H$  to the points in the interior of  $B(\mathbf{x}, \delta')$ . So no ray from the origin can pass through both  $B(\mathbf{x}, \delta')$  and  $B(\mathbf{y}, \delta')$ , as required. Finally, it is easy to see that no larger value of  $\delta'$  can have this property, for when  $\delta' > \sigma_{\mathbf{x}}^2$  we find that the origin is in the interior of  $B(\mathbf{x}, \delta')$ , and so all rays from the origin (including, for example,  $R_{\zeta(\mathbf{y})}$ ) pass through  $B(\mathbf{x}, \delta')$ .  $\square$

**Theorem 4.** *Let  $C \subseteq \mathbb{R}^n$  be a finite set of non-constant codewords. Define the minimum distance  $\delta'$  of  $C$  by*

$$\delta' = \min_{\mathbf{x}, \mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}} \delta'(\mathbf{x}, \mathbf{y}).$$

*The word error probability of a maximum likelihood decoder is bounded above by the probability that  $\chi^2(n-1) \geq \delta'/\sigma^2$ , where  $\chi^2(n-1)$  is the chi-squared distribution with  $n-1$  degrees of freedom.*

*Proof.* When a codeword  $\mathbf{x}$  is transmitted, the decoder receives a vector  $\mathbf{r} = a(\mathbf{x} + \boldsymbol{\nu}) + b$  where  $a$  and  $b$  are unknown, and the components of  $\boldsymbol{\nu}$  are normally distributed with mean 0 and standard deviation  $\sigma$ . Let  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$  be an orthonormal basis for  $\mathbb{R}^n$ , with  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{n-1}$  spanning  $Z$ . We may write  $\boldsymbol{\nu} = \boldsymbol{\nu}' + c\mathbf{e}_n$  where

$$\boldsymbol{\nu}' = \nu'_1 \mathbf{e}_1 + \nu'_2 \mathbf{e}_2 + \dots + \nu'_{n-1} \mathbf{e}_{n-1}$$

and where the real numbers  $\nu'_i$  and  $c$  are independent and normally distributed with mean 0 and standard deviation  $\sigma$ . Now  $\|\boldsymbol{\nu}'\|^2/\sigma^2$  is a chi-squared random variable with  $n-1$  degrees of freedom, so  $\|\boldsymbol{\nu}'\|^2 < \delta'$  with probability equal to the probability that  $\chi^2(n-1) \geq \delta'/\sigma^2$ . Assume that  $\|\boldsymbol{\nu}'\|^2 < \delta'$ . It suffices to show that our maximum likelihood decoder returns the codeword  $\mathbf{x}$ .

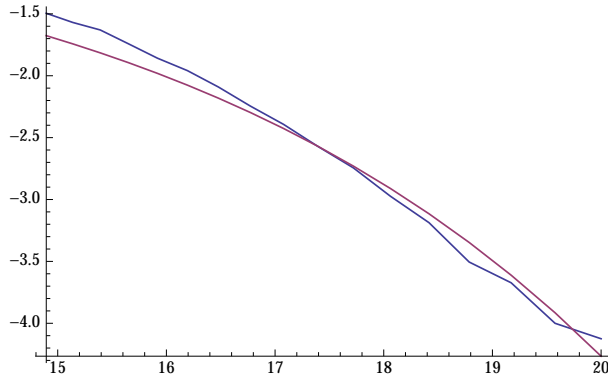


Figure 4: Word error rate when  $q = 2$  and  $n = 4$

Note that  $\zeta(\boldsymbol{\nu}) = \boldsymbol{\nu}'$ , and so  $\zeta(\mathbf{r}) = a(\zeta(\mathbf{x}) + \boldsymbol{\nu}')$ . The ray  $R_{\zeta(\mathbf{r})}$  passes within a squared distance of  $\|\boldsymbol{\nu}'\|^2$  from  $\zeta(\mathbf{x})$ , since  $\zeta(\mathbf{x}) + \boldsymbol{\nu}'$  lies on this ray. So  $\ell_{\mathbf{x}}(\mathbf{r}) \leq \|\boldsymbol{\nu}'\|^2 < \delta'$ . Let  $\hat{\mathbf{x}} \in C$  be such that  $\hat{\mathbf{x}} \neq \mathbf{x}$ . Lemma 3 and the definition of  $\delta'$  shows that  $R_{\zeta(\mathbf{r})}$  cannot intersect the interior of a ball in  $Z$  of radius  $\delta'$  centred at  $\zeta(\hat{\mathbf{x}})$ . Hence  $\ell_{\hat{\mathbf{x}}}(\mathbf{r}) \geq \delta'$ . Thus a maximum likelihood decoder will correctly decode to  $\mathbf{x}$ .  $\square$

## 5 Comparing the two decoders

Simulations indicate that the decoder in [3] has a comparable performance with the maximum likelihood decoder when word error rate is considered. Figure 4 shows the results of a simulation for the maximum likelihood decoder when  $q = 2$  and  $n = 4$  for a range of noise levels when a 1-constrained code [3] is used: the horizontal axis is the signal to noise ratio, defined as  $-20 \log_{10} \sigma$ , and the vertical axis is the word error rate. Each point was the result of 10,000 trials with  $a = 1.07$  and  $b = 0.07$ . Figure 5 gives a similar situation when  $n = 12$ . In each figure, simulation results are plotted along with the error rate predicted by averaging the bound in Theorem 4 over all subcodes of size 2. These parameters are chosen for direct comparison with Figure 5 in [3].

Figure 6 is a scatter plot of two notions of distance for 10,000 random vectors when  $q = 100$  and  $n = 20$ : the distance  $\delta'(\mathbf{u}, \mathbf{v})$  defined in Section 4 and the distance  $d_2(\mathbf{u}, \mathbf{v})$  defined in Section IV.B of [3] for the purposes of

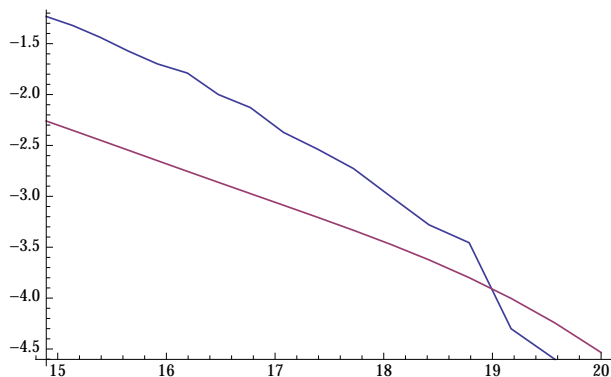


Figure 5: Word error rate when  $q = 2$  and  $n = 12$

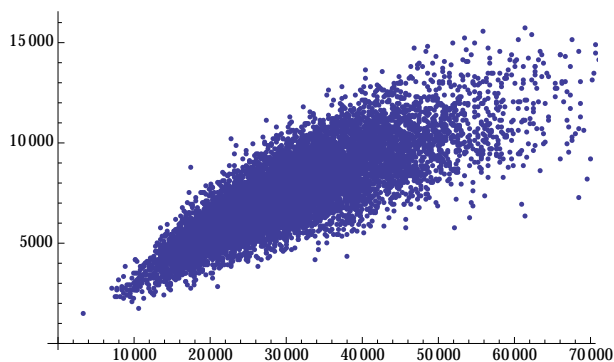


Figure 6: Two distances:  $d_2$  against  $\delta'$

estimating word error rates. The figure shows a close to linear relationship between these two quantities for random vectors. Figure 7 is a scatter plot of two likelihood functions (namely Pearson distance and the likelihood function  $\ell_{\mathbf{x}}(\mathbf{y})$  used by the maximum likelihood decoder) for a similarly randomly generated collection of vectors. Again, a close to linear relationship can be observed, which provides an explanation for the similar performance of the corresponding decoders.

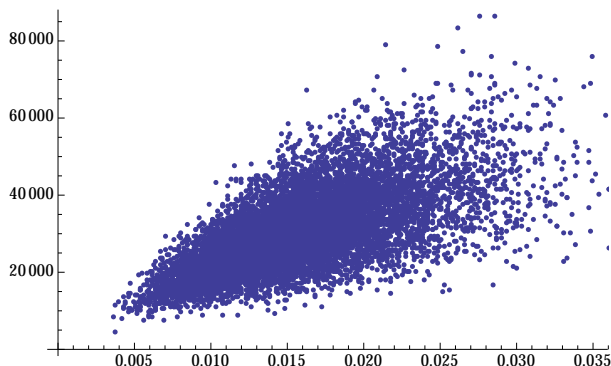


Figure 7: Two likelihood functions: Pearson distance against  $\ell_{\mathbf{x}}(\mathbf{y})$

## 6 Comments

### 6.1 A geometric meaning for Pearson distance

Since the offset  $b$  is arbitrary and unknown, and changes the mean of a vector by  $b$ , it seems sensible to normalise codewords and received words to have mean 0. In other words, we consider the words  $\zeta(\hat{\mathbf{x}}) = \hat{\mathbf{x}} - \bar{\hat{\mathbf{x}}}\mathbf{1}$  and  $\zeta(\mathbf{r}) = \mathbf{r} - \bar{\mathbf{r}}\mathbf{1}$  rather than  $\hat{\mathbf{x}}$  and  $\mathbf{r}$ . Scaling a vector of mean 0 by  $a$  does not change the mean, but scales the standard deviation by a factor of  $a$ . So it seems sensible to scale our normalised vectors so that they have standard deviation 1: if our original vectors were non-constant, we can always find a scaling factor  $a$  that does this. The resulting vectors,  $(\hat{\mathbf{x}} - \bar{\hat{\mathbf{x}}}\mathbf{1})/\sigma_{\hat{\mathbf{x}}}$  and  $(\mathbf{r} - \bar{\mathbf{r}}\mathbf{1})/\sigma_{\mathbf{r}}$ , lie on an  $n - 1$ -dimensional unit sphere, centred at the origin. A natural distance measure between two vectors  $\mathbf{u}$  and  $\mathbf{v}$  on this unit sphere is their squared Euclidean distance, and it is not difficult to show that this is exactly  $\delta_{\text{Pearson}}(\mathbf{u}, \mathbf{v})$ .

### 6.2 Why are constant codewords forbidden?

The (unnormalised) standard deviation of a constant codeword is 0, so the Pearson correlation coefficient  $\rho_{\mathbf{r}, \hat{\mathbf{x}}}$  is not defined when  $\hat{\mathbf{x}}$  is constant. But the forbidding of constant codewords is an artifact of the channel itself, not just the distance measure that is proposed for decoding. To see this, suppose that  $\hat{\mathbf{x}} = \alpha\mathbf{1}$  is a codeword. Let  $\mathbf{r}$  be a received word, and define  $\mathbf{s} = \mathbf{r} - \hat{\mathbf{x}}$ .

For any positive  $\epsilon \in \mathbb{R}$  we have

$$\epsilon^{-1}(\hat{\mathbf{x}} + \epsilon \mathbf{s}) + (1 - \epsilon^{-1})\alpha \mathbf{1} = \mathbf{s} + \alpha \mathbf{1} = \mathbf{r}.$$

Setting  $a = \epsilon^{-1}$ ,  $b = (1 - \epsilon^{-1})\alpha$  and  $\boldsymbol{\nu} = \epsilon \mathbf{s}$  we have that  $\mathbf{r} = a(\hat{\mathbf{x}} + \boldsymbol{\nu}) + b\mathbf{1}$ . But  $\epsilon$  may be taken to be arbitrarily small, and so we see  $\mathbf{r}$  could have been received when  $\hat{\mathbf{x}}$  was transmitted, with an arbitrarily small error vector  $\boldsymbol{\nu} = \epsilon \mathbf{s}$ . So any reasonable decoder for this channel would decode *every* received vector to  $\hat{\mathbf{x}}$ , and a sensible distance measure would set the distance between  $\hat{\mathbf{x}}$  and any other vector as 0.

### 6.3 Future work

Weber, Immink and Blackburn [5] have studied optimal Pearson codes, which are the largest codes contained in  $\{0, q - 1\}^n$  that can be correctly decoded in the zero-error case (when  $\sigma = 0$ , and so  $\nu = \mathbf{0}$ ). It would be very interesting to fully explore the interplay between the error correcting capacity of codes when  $\sigma > 0$  and the rate of an optimal code. We hope that the distance between codewords that is defined in Section 4 will provide a tool to accomplish this.

**Acknowledgements** The author would like to thank Alexey Koloydenko for fruitful discussions on maximum likelihood estimation, and Kees S. Immink and Jos Weber for commenting on an earlier draft of the manuscript. The author would also like to acknowledge the help of two software packages that were used to conduct experiments and simulations: Compass and Ruler [2], and Mathematica [10].

## References

- [1] Roberto Bez, Emilio Camerlenghi, Alberto Modelli and Angelo Visconti, ‘Introduction to flash memory’, *Proc. IEEE* **91** (2003), 489–502.
- [2] René Grothmann, Compass and Ruler dynamic geometry programme, Version 12.0, [http://car.rene-grothmann.de/doc\\_en/](http://car.rene-grothmann.de/doc_en/).
- [3] Kees A. Schouhamer Immink and Jos H. Weber, ‘Minimum Pearson distance detection for multilevel channels with gain and/or offset mismatch’, *IEEE Trans. Information Theory* **60** (2014), 5966–5974.



- [4] Kees A. Schouhamer Immink and Jos H. Weber, ‘Hybrid Minimum Pearson and Euclidean Distance Detection’, *IEEE Trans. Communications*, to appear.
- [5] Jos H. Weber, Kees A. Schouhamer Immink and Simon R. Blackburn, ‘Pearson codes’, preprint.
- [6] Anxiao (Andrew) Jiang, Robert Mateescu, Moshe Schwartz and Jehoshua Bruck, ‘Rank modulation for flash memories’, *IEEE Trans. Information Theory* **55** (2009), 2659–2673.
- [7] Anxiao (Andrew) Jiang, Moshe Schwartz and Jehoshua Bruck, ‘Error-correcting codes for rank modulation’, in *Proc. IEEE Int. Symposium on Information Theory (ISIT)* (IEEE, 2008), 1736–1740.
- [8] Anxiao (Andrew) Jiang, Moshe Schwartz and Jehoshua Bruck, ‘Error-correcting codes for rank modulation’, *IEEE Trans. Information Theory* **56** (2010), 2112–2120.
- [9] Frederic Sala, Kees A. Schouhamer Immink, and Lara Dolecek, ‘Error control schemes for modern flash memories’, *IEEE Consumer Electronics Magazine*, January 2015, 66–73.
- [10] Wolfram Research, Inc., Mathematica, Version 9.0, Champaign, IL (2012).